

情報処理教育のためのポータブルな開発環境

湊 信吾*

Portable Development Environment for Information Processing Education

Shingo MINATO¹

¹ *Department of Information Management, Faculty of Business Administration,
Ishinomaki Senshu University, Miyagi 986-8580, Japan*

Abstract

Computer system for education of Ishinomaki Senshu University runs on thin client system. This system is stable for security, maintenance, and operation, but when using libraries for programming languages such as python and R and new software of blockchain and Ethereum, implementation of these software is delayed until system maintenance that is conducted two times a year. To avoid this problem, I tried to develop some portable development environment using USB memory.

1. 緒言

近年の情報処理教育において次のような新しい分野が取り上げられるようになってきた。

- ・データ・サイエンス
- ・機械学習
- ・ニューラル・ネットワーク
- ・ブロックチェーンに関わる技術（暗号学的ハッシュ関数、P2P ネットワーク、公開鍵暗号法）
- ・自然言語処理
- ・バイオインフォマティクス

大学の情報教育向けのコンピュータシステムではこのような状況を考慮してプログラミング言語や必要なソフトウェアが実装されている。

また、大学のコンピュータシステムにはシンクライアントシステムが採用されている。中央のサーバで端末用のイメージを用意し、コンピュータ室で端末が起動された時にこのイメージがダウンロードされ OS が起動しソフトウェアを利用できる仕様となっている。このようなシンクライアントシステムは運用面での管理が容易であること、セキュリティ面で保障されるなど利便性の高いものである。

ところで授業を行っていると、先に上げた分野においてソフトウェアやライブラリが急に必要になることがある。シンクライアントシステムにおいてはダウンロードできてもユーザのインストー

ルする権利は限られたものとなり、作業を行うことはできない。ソフトウェアのインストールを行うためには年2回行われるメンテナンスの時期まで待たねばならない。

このような状況に対応する方法として、システムの環境に依存しつつも、ソフトウェアやライブラリを利用するためのポータブルな環境を準備することが可能である。ポータブルな環境においては USB メモリーに必要なソフトウェアをインストールしておき、コンピュータ室の端末で動いている OS に合わせて USB メモリー上のソフトウェアを実行できるようにするものである。

しかし、USB メモリーに実装したポータブルな環境で動作するソフトウェアと同じものがすでにコンピュータシステムに実装されている場合には、コンピュータシステム上のソフトウェアが優先的に実行されることを考慮する必要がある。

また、ライブラリをインストールするにあたって大学のコンピュータシステム側のディレクトリにインストールを試みた場合、システム側からインストールは拒否され失敗に終わる。本研究では、このような環境において USB メモリーを使用したポータブルな開発環境の開発、ならびに運用例について報告するものである。

*石巻専修大学経営学部情報マネジメント学科

2. USB メモリーを使用したポータブルな開発環境の開発

ポータブルな開発環境を作るための手順はほぼ同じなので、具体的な例として R と RStudio を用いた開発環境を作る手順を追いながら解説していく。

アメリカの大リーグのデータを使用して打撃や投手成績などの細かい分析を行うための手法としてセイバーメトリクスという手法が確立している。ゼミにおいて R というプログラミング言語を使用してセイバーメトリクスの実験を行おうとした。

大学のコンピュータシステム側にはすでにプログラミング言語 R や R を扱うために利用される統合開発環境 (IDE) として RStudio がインストールされている。通常ならばこのような環境の下で R を利用して基本的な実習を行うことができる。ところがセイバーメトリクスにおいては、コマンドの出力を次のコマンドに渡すためのツールなどが利用できる dplyr やグラフ描画に必要な ggplot2 などのライブラリが必要な事がわかった。

R の library 関数でライブラリの存在を確認したがインストールされていないことがわかった。そこで install.packages コマンドでインストールを試みたところ、ダウンロードまでは進むが最後のインストールの段階においてインストールの権限がないということで、インストールは失敗に終わった。

この状況では実験を行えないし、ライブラリをイメージに取り込んでもらうには 9 月と 3 月のメンテナンス時期まで待つ必要があるが、これでは間に合わない。そこでシステムとは切り離しつつも端末側の OS を利用して必要なソフトウェアやライブラリを実行できるポータブルな開発環境を用意することにした。

ポータブルな開発環境としては常に持ち歩くことができる環境をイメージしてもらいたい。このようなイメージを実現するデバイスとしては USB メモリーや microSD などを持ち上げることができる。これとは別に Docker などのようにネットワーク経由でコンテナと呼ぶ開発環境のイメージを用意して端末にダウンロード後実行する方法も

ある。

今回選択したデバイスには USB メモリーを使用することにした。選択した理由として 2021 年前期において 32GB で比較的低価格でコストパフォーマンスが良いこと、USB のバージョンも 3 以上となっており実習時に支障がないと思われること、microSD に比較して学生が紛失しにくいことが挙げられる。

用意した USB メモリーに R と Rstudio をこの順でインストールする。RStudio は R がすでにインストールしてあることが前提となっているためである。R と RStudio をインストールするために使用したインストーラは次の二つである。

R-4.1.0-win.exe

RStudio-1.4.1717.exe

まず R のインストーラを起動する。画面に従って進んでいくとインストール先を設定する画面になる。ここで図 1 のように USB メモリーのドライブとして e: を頭に付け次に保存用のディレクトリ名を指定する。ドライブ名に e を選んだのは大学のシステム環境に対応させるためである。コンピュータ室の端末前面に USB メモリーを挿すとドライブ名が e に設定される。

起動時オプションはポータブル化する際に邪魔になるので「はい」を選択しカスタマイズする (図 2)。

インストーラはそのままの設定で進めていく。

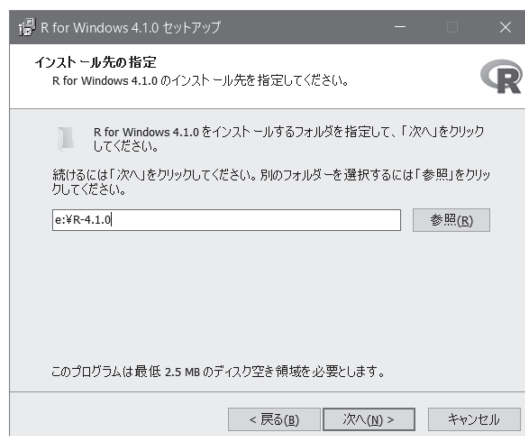


図 1. インストール先の設定 (R)

次の画面で項目が選択されていないことを確認後 (図3) インストールが始まる。

次に Rstudio のインストーラを起動する。図4

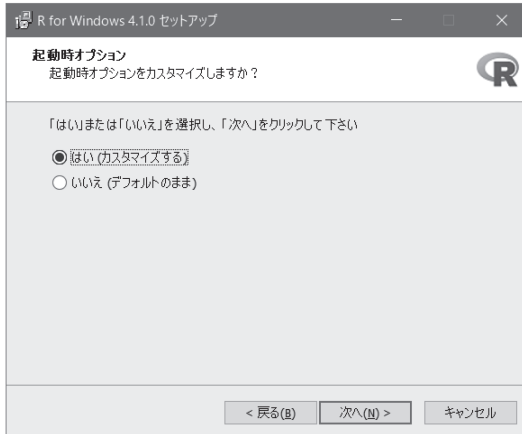


図2. 起動時オプションの選択

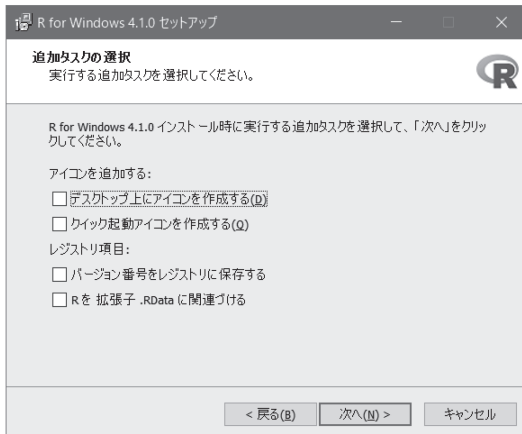


図3. 追加タスクの確認

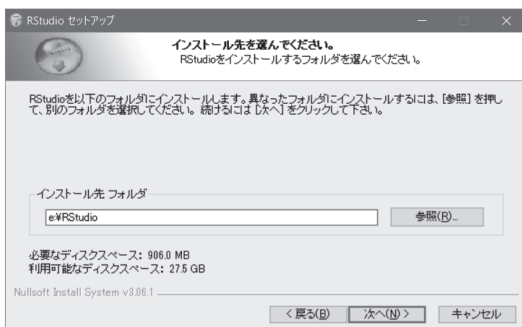


図4. インストール先の設定 (RStudio)

のようにインストール先を聞いてくるので USB メモリーのドライブ (e:¥) のディレクトリ名を指定する。

設定はこれで終了し、インストールを進めてかまわない。

これで USB メモリーに R と RStudio のインストールが完了した。これを大学のシステムと干渉しないように起動するためには、環境変数というものを設定しバッチファイルで起動する必要がある。

適当なテキストエディタで次のプログラムを入力後 r3.bat のようにファイル名を付けて USB メモリーのトップディレクトリに保存する。[.bat] は Windows にバッチファイルであることを示すための拡張子であり、この拡張子によりこのファイルをコマンドプロンプトから起動することができる。

echo Off

```
set PATH=%1:¥R-4.1.0¥bin¥x64; %PATH%
set PATH=%1:¥RStudio¥bin;%PATH%
set R_HOME=%1:¥R-4.1.0
set R_LIBS_USER=%1:¥R-4.1.0¥library
echo on
```

まず大学のコンピュータ室にある端末の前面に USB メモリーを挿すと自動的に e ドライブに設定されてしまう。ところが他のシステムにおいてはドライブ名が変更される可能性があるので、このバッチファイルではドライブ名をコマンドラインのパラメータから受け取り %1 を使って設定している。使い方としては次のようにコマンドラインでバッチファイルに e を指定して実行する。

r3 e

環境変数 PATH に R と RStudio の実行ファイルのある場所を設定している。R_HOME、R_LIBS_USER に R に必要な値を設定する。

エクスプローラーを起動する。図5のように E ドライブを選択後、リボン下のドライブ名を表示する部分をクリックし、cmd と入力しエンターキーをたたく。

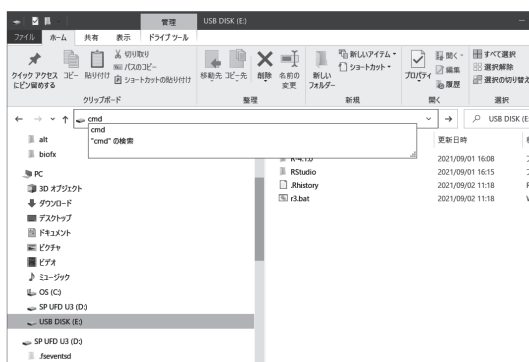


図 5. コマンドプロンプトの起動

コマンドプロンプトが開くので上記に示したコマンドを入力しエンターキーをたたきバッチファイルを実行する。

これで e ドライブにインストールしたソフトウェアを利用することができるようになり、ポータブル化は完成する。

コマンドプロンプトで次のコマンドを入力しエンターキーをたたき R の起動とバージョンを確認する。

```
r --version
```

RStudio も同様に次のコマンドをコマンドプロンプトに入力しエンターキーをたたき起動を確認する。

rstudio

特に左下のペイン (枠) 内に表示されたバージョンを確認しておく必要がある。

以上の準備は自分の PC で行った。ポータブル化が成功したものとなし、実習の前にコンピュータ室で同様の手順で作業を行いテストしてみたところ、表示されたバージョンの違いによりシステム側の R が起動されることがわかった (図 6)。

これは RStudio 側がシステム側の R に関わる値を OS のレジストリに登録し、起動時に参照するようになっているためである。このままでは必要なライブラリを追加しようとしても拒否されてしまう。

レジストリが関わっている場合、毎回起動時に設定を行うことになり面倒ではあるが、Rstudio を起動後、設定を変更することで USB メモリ側の R を起動することができるようになる。

メニューの Tools > Global Options の順にクリックし (図 6)、Options のウィンドウが開くので一番上の R Sessions のすぐ下の R Versions の Change ボタンをクリックする (図 7)。

次のウィンドウで Choose a specific version of R を選択後、Browse ボタンをクリックし E ドライブに設定した R を指定する (図 8)。

ディレクトリを設定するウィンドウが出るので左側のペインで R-4.1.0 > bin > x64 の順にクリックし、ディレクトリを選択する (図 9)。

選択後次のウィンドウが出るので [64bit] E:

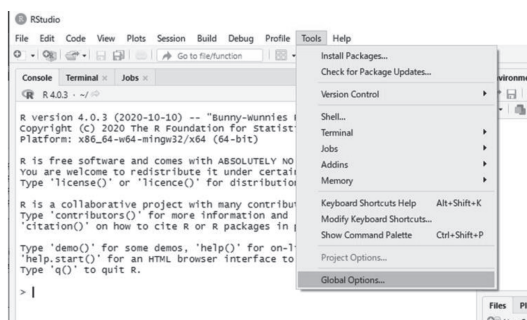


図 6. R のバージョンの確認

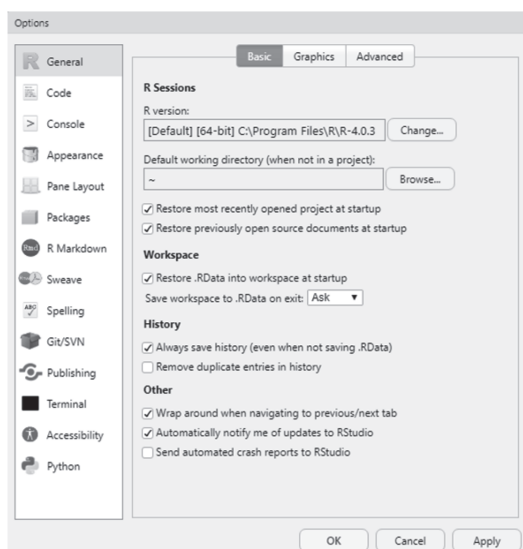


図 7. R の Options ウィンドウ

R-4.1.0 を選択後、OK ボタンをクリックする（図 10）。

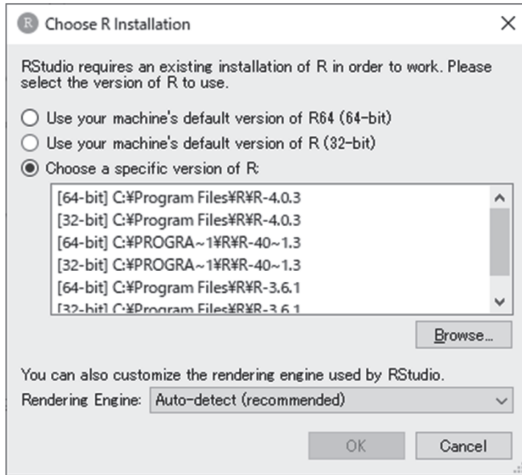


図 8. R のバージョンを設定するウィンドウ



図 9. USB メモリー側の R のディレクトリの選択

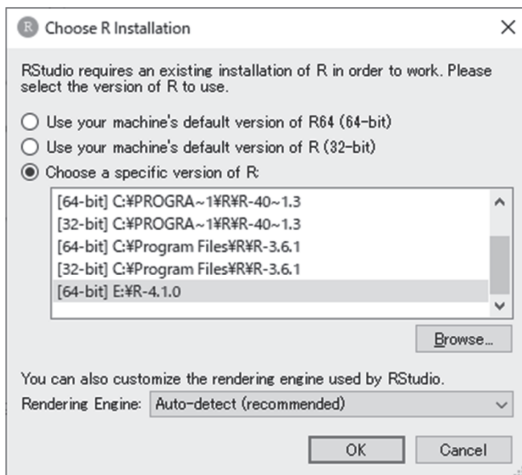


図 10. R のバージョンの選択

設定後、OK ボタンをクリックして進む。再起動するように聞いてきたら Yes ボタンをクリックする。RStudio が再帰動し USB メモリー側の R を利用できるようになる。

以上、大学のシステムにおいて RStudio のようにインストール時にシステム側のレジストリに登録するものについては完全にポータブル化を行うことができないが、設定の変更により対応できる。

また Firefox Portable⁽¹⁾ のようにポータブル化を前提にパッケージを公開している例もある。

通常はポータブル化に対応させるためインストーラとは別に zip ファイルで提供している例も多くある。

この例の他に実習用に次のようなポータブル開発環境を作成した。

① ruby、iruby⁽²⁾、python、jupyter notebook、GR.rb⁽³⁾（グラフ描画用ライブラリ）を同梱し、jupyter notebook で ruby を使えるようにし、グラフも描画できるようにした開発環境。この開発環境は 2021 年 8 月の東松島高校の高大連携授業の実習において使用した。

②ブロックチェーン用の開発環境として python、Geth⁽⁴⁾、node.js⁽⁵⁾、MetaMask⁽⁶⁾、Firefox Portable を同梱して開発環境を用意した。2021 年 9 月の宮城県高校産業研修会において利用する予定であった。

①と②の例においてもバッチファイルを用意し、環境変数を設定しポータブル化を完成させる必要がある。

以下にこの開発環境の長所と短所を挙げておく。

<長所>

- ・USB メモリーを回収することで学生の作業、理解度を確認することができる。特に RStudio、jupyter notebook においてコマンドの履歴や作業過程を追跡することが容易となる。

- ・必要なライブラリをすぐにインストールすることができる。

<短所>

- ・開発環境が大きなサイズとなる場合、複製する

際に時間がかかる。

- ・同様に複製する手間を考えると実習可能な人数は多くとも 30 人が適当のように思う。
- ・USB メモリーにはメーカーおよび同じメーカーであっても読み込み速度、書き込み速度に差があるため、考慮して作業を行う必要がある。

3. まとめ

シンクライアントシステムを採用しているシステムでは急にソフトウェアやライブラリの追加を要求することは難しい。解決法の一つとして USB メモリーを利用したポータブルな開発環境を作成できる例を紹介した。

4. 文献リスト

- (1) https://portableapps.com/apps/internet/firefox_portable
ブラウザソフト Firefox をポータブルな環境で扱えるようにしたものを提供している。
- (2) <https://github.com/SciRuby/iruby>
jupyter プロジェクトで ruby を利用可能にするため

のカーネルと呼ばれるソフトウェア

- (3) <https://github.com/red-data-tools/GR.rb>
ruby 用の視覚化用ライブラリ。グラフ作成もできる。グラフを ruby の簡単なプログラムで表現できる。
- (4) <https://geth.ethereum.org/downloads/>
Go Ethereum で配布されている Ethereum のためのクライアントソフト。Ethereum はブロックチェーンの技術を利用した暗号通貨や DApp と呼ばれるソフトウェアを作成するためのプラットフォーム（技術基盤）。
- (5) <https://nodejs.org/ja/>
JavaScript をベースにした開発環境。npm というツールを利用することで、様々なソフトウェアをインストールすることができる。
- (6) <https://metamask.io/>
暗号通貨の取引で Wallet と呼ばれる暗号鍵を管理するためのツールとして有名。Chrome での使用例が多くみられるが多くのブラウザの extension（拡張機能）に対応している。